

Server-based multi-standard home network bridging

This application is a Continuation-in-Part of U.S. serial no. 09/340,272 (attorney docket PHA 23,634) filed 6/25/99 for Yevgeniy Eugene Shteyn for BRIDGING MULTIPLE HOME NETWORK SOFTWARE ARCHITECTURES.

5 The invention relates to the bridging of multiple networks based on different software architectures. The invention relates in particular to home networking.

It seems to be unlikely that there will ever be a single universally applicable networking standard for each device in the home. Multiple standards for software architectures coexist and new ones will emerge. New standard interfaces will be developed for new types of devices that are specifically targeted by those standards. Home applications
10 are designed to make an intelligent use of all devices available in the home, but are not capable of dealing with each and every networking standard in use or becoming available in the future. Similarly, devices themselves will not be able to support every existing home networking standard. For these reasons, bridges are needed between the different sub-networks or clusters, each respective one complying with a specific respective standard. A
15 bridge serves to transparently represent a device, complying with a first standard in a network cluster of a first type, as a device complying with a second standard in a network cluster of a second type. The result is a single unified view of the home network available to software applications written for the first standard and as well those written for the second standard.

Typically, a device on a network is controlled through a set of messages
20 complying with the interface of the device. Interoperability between devices and software applications depends upon standard interfaces with a unique identification. Once an application knows the unique identification of the device, it knows the interface of the device and it can control the device by sending it messages. To the application it makes no difference whether it is sending standard messages directly to the device itself, or indirectly
25 via a software component that translates these messages into a different set of message that (eventually) achieve the same desired effect or state change at the controlled device. Hence, a bridge between networks of multiple different standards can be considered a multi-standard device. That is, the bridge complies with each of these multiple standards, and hosts software components that translate device interfaces from a first to a second standard and vice versa.

For example, a bridge is used between a cluster of five devices complying with a standard A and three devices complying with a standard B, different from A. The bridge hosts three translation modules for translating from A to B and five translation modules for translating from B to A. Accordingly, a software application capable of interacting with only standard A devices can now control all eight devices.

U.S. serial no. 09/340,272 (attorney docket PHA 23,634) filed 6/25/99 for Yevgeniy Eugene Shteyn for BRIDGING MULTIPLE HOME NETWORK SOFTWARE ARCHITECTURES, herein incorporated by reference, relates to the bridging of home networks of different software architectures. References to software representations of devices and services on a first one of the networks are automatically created. The references are semantically sufficient to enable automatic creation of at least partly functionally equivalent software representations for a second one of the networks so as to make the devices and services of the first network accessible from the second network. This document also addresses the HAVi, Home API and Jini software architectures.

In the following, the expression "translation module" includes the concept of "software representation", i.e., the representation in software of a physical device or of a service on a network or a sub-set thereof so as to make the device or service accessible to the relevant messaging or controlling software.

Preferably, a bridge performs the following functions: detection of the addition of a device in either of the bridged networks; identification of the type of the added device; locating the translation module for the identified device type if the device is likely to be of interest to the other network; and installing the translation module on the other network according to the procedure required by the standard used by that network.

Successful standards will continue to define standard interfaces for newly developed devices in a domain considered relevant to those standards. This necessitates the development of accompanying translation modules that enable those new devices to be represented in networks based on different standards. As a consequence, a bridge is unable to contain all embedded translation modules for all possible, relevant devices that become available in the future.

Accordingly, the inventors propose a solution wherein a bridge is connected to a server, e.g., on the Internet. This server offers a lookup service for some set of standards, and allows a bridge to locate and download the appropriate translation modules for use in the home network.

More specifically, the invention relates to a method of providing a service to a user of a home network. The method comprises enabling a component of a first cluster in the network to interact with a second cluster of the home network. The first cluster has a first software architecture, and the second cluster has a second software architectures different from the first one. The first and second clusters are coupled through a bridge. The method comprises enabling a server external to the clusters, e.g., on the Internet, to receive a reference of a component having a first software representation in the first cluster. The bridge may provide this reference. The method further comprises enabling to provide to the bridge a translation module, associated with the reference, for at least partially representing the component on the second cluster upon the module being installed on the bridge.

A service provider thus can maintain and update a data base of translation modules for any multiple standards being used in home networks. This partitioning and delegation of functionalities has many advantages as discussed below.

The invention allows the bridge to be fairly "light-weight" or low-cost, as it does not need to have embedded translation modules for all possible devices of all standards that it could be connected to in the home. Storage and computation power is only needed for the devices that are actually bridged (i.e., storage is not needed for the ones that are *potentially* bridged), and only *when* they are bridged (e.g., "just-in-time"-bridging, i.e., at the time of connecting the new device to the network).

Further, the invention renders the home network as a whole extensible and future-proof. As new devices are invented and descriptions thereof become part of various standard specs, these descriptions are translated and uploaded to the bridge server by, e.g., the device manufacturer or a third party, to make them available for bridging in existing home networks. This process does not require any update mechanism of components in the home network itself.

A further benefit is that the bridge-server operator is able to obtain information about the configuration of the individual user's home network. This information can be used to the advance of both the user and the manufacturers and service providers. See, for example, U.S. Serial No. 09/160,490 (attorney docket PHA 23,500) filed 9/25/1998 for Adrian Turner et al., for CUSTOMIZED UPGRADING OF INTERNET-ENABLED DEVICES BASED ON USER-PROFILE, incorporated herein by reference. This document relates to a server system that maintains a user profile of a particular end-user of consumer electronics network-enabled equipment and a data base of new technical features for this type of equipment, e.g., a home network. If there is a match between the user-profile and a new

technical feature, and the user indicates to receive information about updates or sales offers, the user gets notified via the network of the option to obtain the feature. Also see, e.g., U.S. Serial No.09/189,535 (attorney docket PHA 23,527) filed 11/10/98 for Yevgeniy Shteyn for UPGRADING OF SYNERGETIC ASPECTS OF HOME NETWORKS, incorporated herein
5 by reference. This document relates to a system with a server that has access to an inventory of devices and capabilities on a user's home network. The inventory is, for example, a look-up service as provided by HAVi, Jini and Home API architectures. The server has also access to a data base with information of features for a network. The server determines if the synergy of the apparatus present on the user's network can be enhanced based on the listing
10 of the inventory and on the user's profile. If there are features that are relevant to the synergy, based on these criteria, the user gets notified. In this sense, U.S. serial no. 09/189,535 relates to the concept of an "application suggestor".

A further advantage of the invention resides in the fact that the server operator is enabled to measure market demands for particular devices to be bridged to a specific
15 standard. The device manufacturer or another relevant third party can be notified of the emerging demand. When the new translation modules become available on the server, bridges that have sent requests for translation modules in the past with which the server could not comply, can now be notified of an upgrade.

Note that the bridge can be implemented as a software component on a device
20 of a specific cluster on the home network in order to provide a bridge to another cluster. For example, a HAVi set top box can have a software component for bridging the HAVi cluster to, e.g., a UPnP cluster on the network. Similarly, a PC that controls the UPnP cluster can have a software component that serves to bridge the UPnP cluster of the home network to the HAVi cluster.

25

The invention is explained in further detail, and by way of example, with reference to the accompanying drawing, wherein:

Fig.1 is a block diagram illustrating the principle of the bridging between two
30 networks according to the invention;

Fig.2 is a block diagram illustrating bridging HAVi to UPnP; and

Fig.3 is a block diagram illustrating bridging UPnP to HAVi.

Throughout the drawing, same reference numerals indicate similar or corresponding features.

As mentioned above an aspect of the invention relates to connecting a bridge to a server, e.g., on the Internet. This server offers a lookup service for some set of standards, and allows a bridge to locate and download the appropriate translation modules in the home network that eventually enables a device on a sub-network of a first architecture to work with devices on a sub-network of a second architecture.

Fig.1 is a diagram of a home network system 100 with a first cluster 102 of devices 104, 106 and 108 that comply with a first software architecture standard, herein-after called standard A. System 100 comprises a second cluster 110 of devices 112, 114 and 116 that comply with a second software architecture standard, herein-after called standard B. Clusters 102 and 110 are interconnected through a bridge 118. In order to have a meaningful network interaction between cluster 102 of standard A on the one hand, and cluster 110 of standard B on the other hand, translation modules are introduced. These modules need to participate in both clusters 102 and 110. The modules typically need components local to the clusters, such as lower-level communication software, in order to be capable of this participating. Rather than having each translation module comprising its own communication software it is more efficient to provide this software as an element of platform components 120 of bridge 118.

The process of the invention is now illustrated with an example wherein B-device 116 is going to be added to system 100.

The first step comprises physically connecting B-device 116 to B-cluster 110, or "booting" B-device 116.

In a next step, bridge 118 detects B-device 116 as a new addition, either because bridge 118 scans B-cluster 110 or its registry/directory/look-up service (not shown) periodically or because B-cluster 110 actively notifies bridge 118. Bridge 118 comprises a software component 122, referred to as Installation Manager, that handles the installation of further software components needed to integrate B-device 116 into system 100. An aspect hereof is discussed in, e.g., U.S. serial no. 09/340,272 (attorney docket PHA 23,634). In the latter document a software component, referred to as the Reference Factory, is capable of extracting information from any of the software representations of devices registered. This Reference Factory is capable of querying the inventory of services or of getting notified of a new software representation according to the methods of the relevant software architecture. Similarly, Installation Manager 122 receives or retrieves information descriptive of newly

added B-device 116. The descriptive information is possibly reformatted before being sent to a bridge server 124 via the Internet 126. In addition, bridge 118 preferably provides information about the local execution environment of home network 100. This information is relevant to the software components that server 124 downloads onto bridge 118. The relevant information regarding the environment relates to the software architectures present, in this case the A-standard cluster 102 and the B-standard cluster 110. The information may also relate to the memory available, the type of operating system(s) being used, virtual machines present, platform libraries, etc., on bridge 118. Based on this information server 124 is able to select the proper translation module or modules that fits or fit in best with the network environment of system 100.

Upon receipt of the descriptive and environment information, server 124 uses a look-up service that needs to match the information, descriptive of B-device 116, with a translation module for representing B-device 116 in A-cluster 102. In general, server 124 has a plurality of look-up services available: one for each ordered pair (X,Y), wherein X and Y are the standards supported by server 124. In order to support bi-directional bridging between a cluster of standard X and another cluster of standard Y, two look-up services are needed: (X,Y) and (Y,X). For support of bi-directional bridging between three clusters, all with different standards P, Q and R, six look-up services are needed: (P,Q); (Q,P); (P,R); (R,P); (Q,R); and (R,Q). Of course, server 124 may only support uni-directional bridging.

Devices, such as devices 104-108 and 112-116, are often composite objects. For example, a TV set typically comprises Display, Amplifier and Tuner components. Server 124 could first try to translate the composite object as a whole into a new composite device with an equivalent functionality. If that does not succeed server 124 could translate as much subsidiary components as possible on a one-to-one basis. This then would result in a partial, but still useful, mapping. For example, if A-cluster 102 has not available a definition of a Tuner, a B-type TV could still be bridged to A-cluster 102 as a monitor-like display/amplifier device. If a one-to-one relationship does not exist between particular subsidiary components in standard A and standard B, then a one-to-many or many-to-many mapping could be used. For example, standard A might define volume control and equalizer effects as a single subsidiary component, whereas standard B distinguishes them as separate subsidiary components. In this case, a device in B-cluster 110 that contains just the volume control component, but not the equalizer component, cannot be bridged to A-cluster 102. On the other hand, a device in A-cluster 102 that contains a single amplifier component (volume control as well as equalizer) can be bridged to network B by applying a one-to-two mapping

of the subsidiary components. In the most general case, a particular set of subsidiary components in A-cluster 102 matches with another set of subsidiary components in B-cluster 110 under a many-to-many mapping.

Next, assume that a matching translation module 128 has been found it is downloaded to the bridge, installed on platform 120 and registered in accordance with the protocol of standard A. This enables other applications and devices of A-cluster 102 to discover and use device 116 through module 128. The installation and registering of module 128 may be postponed until after it has been run on the execution environment of bridge 118.

The invention is explained below with an example illustrating the bridging of HAVi and Universal Plug and Play (UPnP) home networks with reference to Figs.2 and 3. The HAVi, Home API, and Jini standards for software architectures in the home networking field have been discussed to some detail in U.S. serial no. 09/340,272 (attorney docket PHA 23,634) mentioned above and incorporated herein by reference. In HAVi, a DCM (Device Control Module) is a software element that represents a single device or functionality on the HAVi network. The DCM exposes the HAVi defined APIs for that device. DCMs are dynamic in nature: if a device is inserted or removed from the network, a DCM for that device needs to be installed or removed, respectively, in the network. DCMs are central to the HAVi concept and the source of flexibility in accommodating new devices and features into the HAVi network.

Universal Plug and Play (UPnP) is an open network architecture that is designed to enable simple, ad hoc communication among distributed devices and software applications from multiple vendors. UPnP leverages Internet technology and extends it for use in non-supervised home networks. UPnP aims at controlling home appliances, including home automation, audio/video, printers, smart phones, etc. UPnP distinguishes between Control Points (CPs) and controlled devices (CDs). CPs comprise, e.g., browsers running on PCs, wireless pads, etc., that enable a user to access the functionality provided by controlled devices.

UPnP defines protocols for discovery and control of devices by CPs. UPnP does not define a streaming mechanism for use by Audio/Video devices. Some of the discovery and control protocols are part of the UPnP specification while others are separately standardized by the IETF (Internet Engineering Task Force). Interaction between CPs and devices is based on the Internet protocol (IP). However, UPnP allows non-IP devices to be proxied by a software component running on IP-compliant devices. Such a component, called

Controlled Device (CD) proxy, is responsible for translation and forwarding of UPnP interactions to the proxied device.

5 A UPnP device has a hierarchy of sub-devices with at the lowest level services. Both devices and services have standardized types. A device type determines the sub-devices or services that it is allowed to contain. A service type defines the actions and state variables that a service is allowed to contain. State variables model the state of the device, actions can be invoked by a CP in order to change that state. The description of the state variables and the action is called the SCP (Service Control Protocol). A UPnP device provides a description of itself in the form of an XML document. This document contains, among other things, the service types that it supports. Optionally, a device may have a presentation server for direct UI control by a CP.

UPnP relies on AutoIP, which provides a means for an IP device to get a unique address in the absence of a DHCP server. UPnP defines a discovery protocol, based on UDP multicast, called SSDP (Simple Service Discovery Protocol). SSDP is based on devices periodically multicasting announcements of the services that they provide. An announcement contains a URL to which service actions are to be sent: the control server. In addition to that, CPs may query the UPnP network for particular device or services types or instances.

UPnP relies on GENA (Generic Event Notification Architecture) to define a state variable subscription and change notification mechanism based on TCP.

After a CP has detected a service it wants to use (via SSDP), it controls the service by sending SCP actions to the control server URL or querying for state variables. Actions are sent using HTTP POST messages. The body of these messages are defined by the SOAP (Simple Object Access Protocol) standard. SOAP defines a remote procedure call mechanism based on XML.

The translation modules, or software representations, of HAVi devices in UPnP are called Controlled Device (CD) *proxies*, while the software representations of UPnP devices in HAVi are called *Device Control Modules (DCMs)*.

HAVi to UPnP bridging

30 Fig 2 is a block diagram of a home network system 200 illustrating the bridging from HAVi to UPnP, and shows in bold arrows the sequence of steps to bridge a HAVi device, here a digital camera, to UPnP.

System 200 has a UPnP cluster formed by devices 202, 204, and 206. Device 202 comprises a light, device 204 comprises an MP3 player, and device 206 comprises a

printer. System 200 has a HAVi network cluster formed by a TV 208, and a digital video recorder 210. The clusters are connected through bridge 118.

In a step 212 a HAVi Camera 214 is physically plugged into the HAVi's 1394 network, thus making Camera device 214 an active HAVi node.

5 In a step 216 this addition is discovered by the HAVi Event Manager, residing on platform components group 218. The HAVi platform listens and reacts to the HAVi NewSoftwareElement event, or listens to a HAVi NetworkReset event to discover Camera 214 as new device.

10 In a step 220 the registration attributes of the DCM of Camera 214 and its FCM components are retrieved from the HAVi Registry on platform 218, and are encoded in some format understood by a bridge server 222, for example XML, and are sent to bridge server 222 using HTTP POST. Bridge 118 may use a HAVi Webproxy FCM to implement this.

15 In a step 224 a look-up component maps a HAVi device description, in the form of DCM/FCM registration attributes, to a UPnP CD proxy 226 for that device. Since UPnP CD proxies and HAVi DCMs are composite objects, the location process might be implemented on the sub device (component) level, as described earlier. In HAVi a device (software representation is DCM) consists of a number of *functional components* (software representation is FCM). To find out which FCMs are part of a DCM bridge 118 can use the 20 DCM::GetFcmSeidList and FCM::GetFcmType methods, or look at registry entries that have the same values for the GUID and n1 field of the TargetId attribute. In UPnP, a device is a hierarchical structure of sub devices with at the lowest level *services*. FCMs serve the same purpose as services. The mapping of HAVi device to UPnP CD proxy 226 might be from complete DCM to complete CD proxy, or partially from FCMs to proxy services. The 25 mapping of FCM to service might be 1-to-1, 1-to-many or many-to-many.

In a step 228 downloaded CD proxy 226 is run on the execution environment of bridge 118. This involves installing an http server for the unique URL of CD proxy 226.

In a step 230 CD proxy 226 sends out periodic announcement messages, and responds to discover messages. This enables the other UPnP applications and devices to 30 discover and use the HAVi Camera 214 through the CD proxy 226.

UPnP to HAVi bridging

Fig.3 illustrates the steps to bridge a UPnP device, here printer 206 to HAVi cluster 208, 210, 214 in system 200.

In a step 302, UPnP Printer 206 is physically plugged into the UPnP network, and 'powering-up' the UPnP device.

A next step 304 involves listening and reacting on the UPnP device announcement message.

5 In a step 306, the device description document of printer 206 is retrieved from the URL embodied in the announcement message, and the document is sent to bridge server 222 using HTTP POST.

A step 308 involves a look-up component that maps a UPnP device description, in the form of a description document (in XML), to a HAVi DCM for that
10 device, here printer 206. Since UPnP CD proxies and HAVi DCMs are composite objects, the location process might be implemented on the sub device (component) level, as described earlier. In HAVi a device (software representation is DCM) consists of a number of *functional components* (software representation is FCM). In UPnP, a device is a hierarchical structure of sub devices with, at the lowest level, *services*. Services that are part of a UPnP
15 device can be found in the device description document. FCMs serve the same purpose as services. The mapping of HAVi device to UPnP CD proxy might be from complete DCM to complete CD proxy, or partially from FCMs to services. The mapping of FCM to service might be 1-to-1, 1-to-many or many-to-many.

A step 310 involves running a downloaded Printer DCM 312 in the execution
20 environment of bridge 118. This involves calling the DCM's Install method.

A step 314 involves DCM 312 and its FCMs to create HAVi software elements, and using those to register with the HAVi registry component (which is part of platform components 218 available on bridge 118).

In a step 316 the HAVi registry posts global NewSoftwareElement events for
25 DCM 312 and all FCMs that are part of it. This enables the other HAVi applications and devices to discover and use UPnP Printer 206 through Printer DCM 312.

CLAIMS:

1. A method of providing a service to a user of a home network (100), wherein:
 - the method comprises enabling a component (116) of a first cluster (110) in the network to interact with a second cluster (102) of the home network;
 - the first cluster has a first software architecture;
 - 5 - the second cluster has a second software architectures different from the first architecture;
 - the first and second clusters are coupled through a bridge (118);
 - the method comprises:
 - enabling a server (124) external to the clusters to receive a reference of a
 - 10 component having a first software representation in the first cluster; and
 - enabling to provide to the bridge a translation module (128), associated with the reference, for at least partially representing the component on the second cluster upon the module being installed on the bridge.
- 15 2. The method of claim 1, wherein the server receives the reference from the bridge.
3. The method of claim 1, wherein the bridge contacts the server via the Internet.
- 20 4. The method of claim 1, wherein the first cluster comprises a HAVi cluster.
5. The method of claim 1, wherein the first cluster comprises a UPnP cluster.
6. A data base comprising at least one translation module (128) for enabling a
- 25 component (116) on a first home network cluster (110) of a first software architecture to interact with a second home network cluster (102) of a second software architecture upon the module having been downloaded via the Internet on a bridge (118) that couples the first and second clusters.

1/3

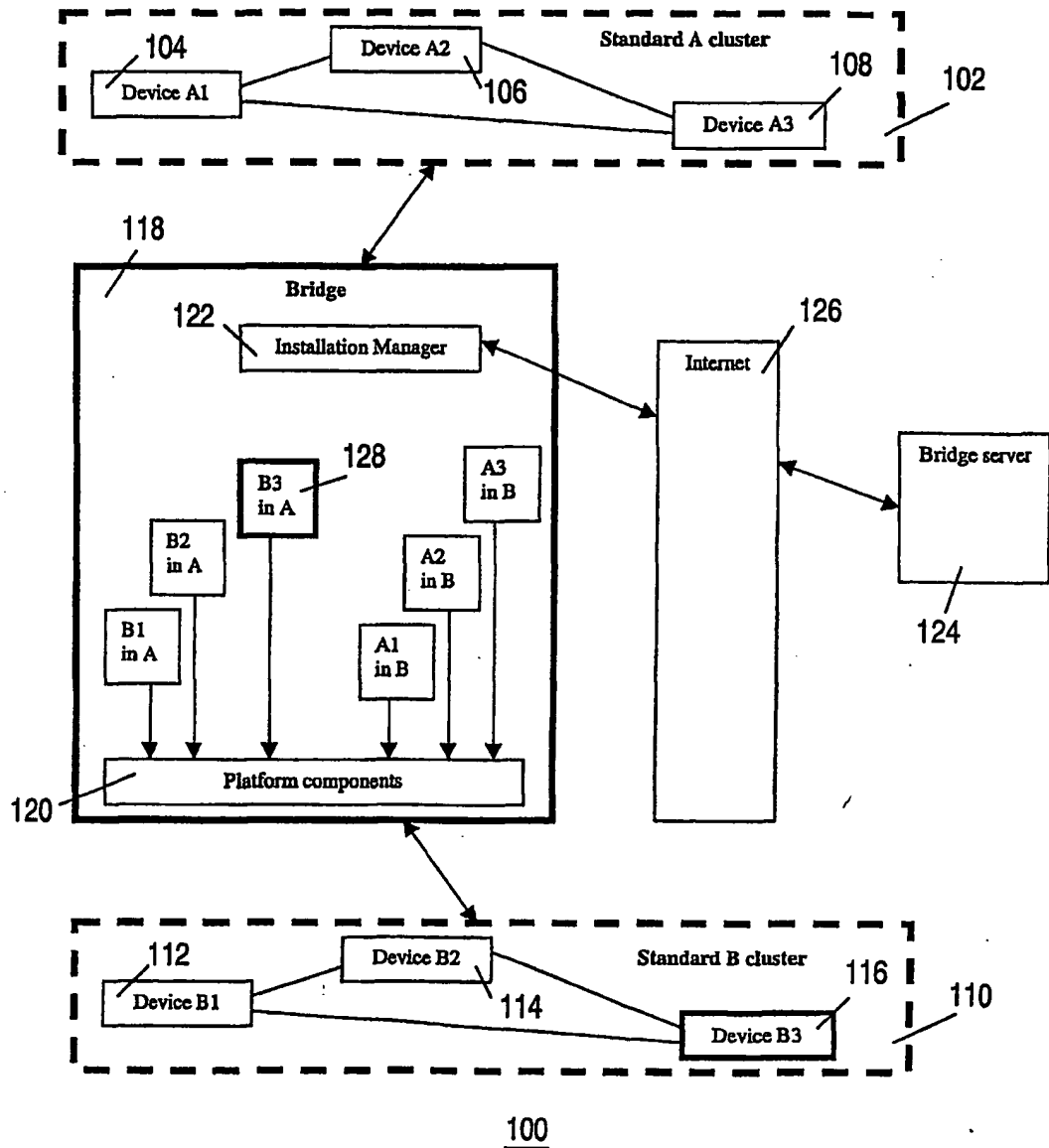


FIG. 1

2/3

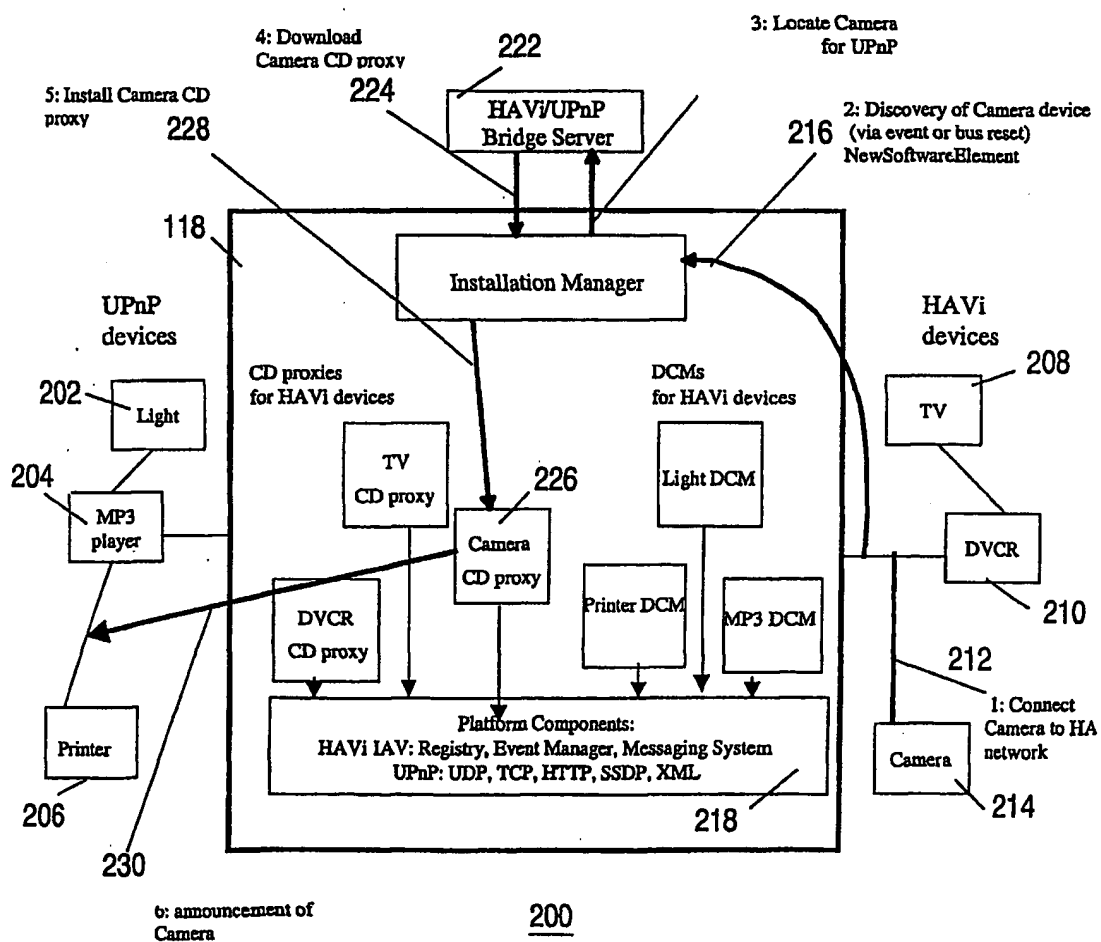


FIG. 2

3/3

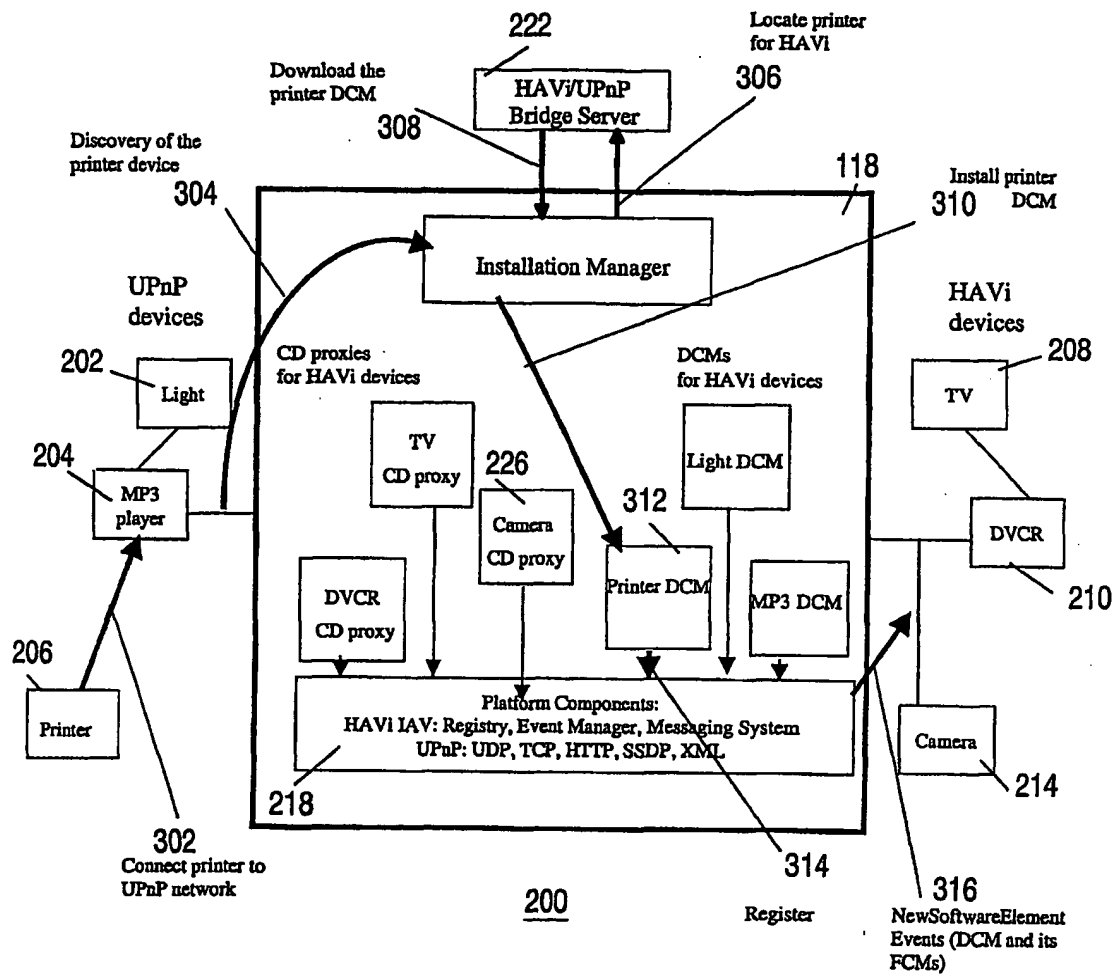


FIG. 3

信息家电与智能家庭网络

·综述·

蒋小洛, 朱海涛

(温州大学 信息科学与工程学院, 浙江 温州 325027)

【摘要】信息家电的出现和家庭网络技术的发展,促进了智能化、网络化家庭的实现。介绍了信息家电和家庭网络的概念、功能以及智能家庭网络的结构,指出了智能家庭网络中存在的主要问题,最后提出了智能家庭网络的控制平台和 OSGi 服务体系结构。

【关键词】信息家电; 智能家庭网络; 控制平台; 开放式服务网关

【Abstract】 The emergence of information appliances, the development of home network technology, hasten the implement of intelligent and networking home. In this paper, the conception and function of information appliance and home network, and the architecture of intelligent network are introduced. The main problems in intelligent home network are also pointed out. At last, the control platform of intelligent home network and the architecture of OSGi service framework are put forward.

【Key words】 information appliance; intelligent home network; control platform; open service gateway initiative

1 引言

随着控制、通信、计算机和网络等技术的迅速发展,人们在家里就可以进行收发 E-mail、网上购物、接受远程教育、远程医疗、……。根据用户需要随时随地监测安全系统(如盗窃、火灾、煤气泄漏等)和控制家用电器(如空调、冰箱、音像设备等)。于是信息家电与家庭网络相伴而生。

2 家庭网络的概念和功能

2.1 家庭网络的概念

家庭网络指的是在一个家居中建立一个通信网络,将家电和设备互相连接起来,实现对所有家庭网络上的家电设备的远程使用和控制及任何要求的信息交换,如音乐、电视或数据等。

2.2 智能家庭网络实现的主要功能

(1)家庭自动化。能够实现对家用电器设备的自动化控制和远程控制功能,对电表、水表和煤气表用量进行自动数据采集、计量,并将采集结果传送给小区物业管理系统。

(2)家庭安全防范。当发生盗窃、火灾、煤气泄漏等时,通过电话线路或互联网传送到报警中心,或拨打用户主人手机、传呼机。当遇到意外情况(如疾病或有人非法侵入)发生时,通过报警按钮向小区物业

管理中心进行紧急呼救。

(3)家庭通信与网络应用。传递数字与多媒体信息,如电话、传真、计算机等,包括 Web 浏览器、收发 E-mail、聊天、网上购物等等。

(4)娱乐。连接各种娱乐性家用电器,如高清晰度电视机、DVD、家庭影院等。

3 信息家电的发展

近年来因特网发展迅速,1996 年全球互联网用户为 4 000 万,1999 年增加到 26 000 万,预计到 2005 年将增加到 10 亿。因特网走进了人们的工作和生活中,成为信息社会中很重要的组成部分,人们开始注意到技术如何与家居生活更贴近的问题。在这种背景下,1999 年底,提出了“信息家电(Information Appliance)”的概念,以网络为主要特点的信息家电成了人们关注的焦点。

3.1 信息家电的概念

信息家电是指将数字化技术引入家用电器领域,用以接收、发布、处理信息,使之成为网络终端或信息处理终端;从广义上来讲包括机顶盒(STB)、WebTV、PDA、掌上电脑、视频游戏设备等所有能提供信息服务或通过网络系统交互信息的消费类电子产品。它具有如下几个基本特点:网络化、数字化、信息化、智能化、成本低、稳定性强等特点。

3.2 嵌入式系统的发展

嵌入式系统是信息家电的核心。嵌入式系统的发展促进了信息家电的研制与开发,掌上电脑、PDA、电脑手机、电脑汽车等新产品层出不穷。由于它对系统所需配置要求低、实时性强、使用灵活、性能价格比高,能够嵌入到现有任何信息家电和工业控制系统中。

3.3 家用电器的上网

家电自身的发展要实现智能化,与外界的沟通要实现网络化,家用电器的上网给信息家电提供了必要条件,目前主要有以下几种方案。

(1)在家用电器内部控制器上嵌入支持 TCP/IP 协议的芯片,直接上互联网,如图 1 所示。

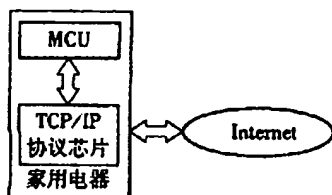


图1 家电上网方案一

(2)在家用电器中加入 WebChip™ 芯片,片内驻留有 MCUnet™ 协议,和 OSGi 协议兼容,使之通过 Webchip™ 网络芯片与 Gateway 连接,再进入因特网,如图 2 所示。

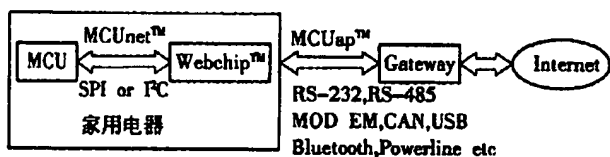


图2 家电上网方案二

这种方案将复杂的因特网协议的处理任务转移到了网关,降低了家用电器与因特网连接对 MCU 性能的要求。

(3)利用 EmWare 公司的 EMIT 技术,并在家用电器中采用 emNet 协议,再通过 emGateway 和因特网连接,如图 3 所示。

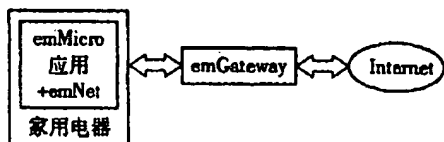


图3 家电上网方案三

(4)在家用电器中嵌入 Neuron Chip,利用 Lonworks 组成局域网,再通过网关和因特网连接,如图 4 所示。

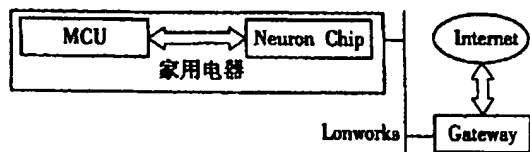


图4 家电上网方案四

(5)采用 32 位高档单片机,在 RTOS(实时多任务操作系统)平台上进行软件开发,在嵌入式系统中实现 TCP/IP 协议处理。

信息家电带来了两方面的优点,一方面,用户可以通过因特网对家电进行远程监视和控制;另一方面,家电制造商可通过因特网对售出的家电产品进行智能化维护。

4 智能家庭网络的实现

4.1 智能家庭网络结构

在家庭网络系统中,一般采用分布式结构。分布式网络结构能实现各节点信息共享,节点之间互相平等。任意两个节点之间都可以互相通信,一旦节点出现故障,只要拿走该故障节点即可,不会影响系统的运行。作为一个智能家庭网络,主要构建家庭控制系统、家庭娱乐系统、家庭数字通信系统三种子网络,具体如图 5 所示。

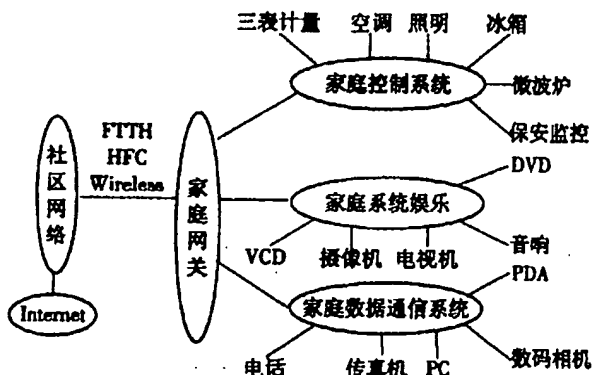


图5 智能家庭网络结构图

4.2 目前可用于构建家庭网络的软硬件技术

目前家庭网络技术分为有线和无线两大类,传输介质可利用家里已有布线系统,如电话线、电力线、同轴电缆、结构化布线(双绞线与光纤),或无线方式。通信标准协议、方案及规范目前很多,有线系

统包括电力线标准 X-10、CEBus、现场总线技术 LonWorks、电话线联网 HomePNA、家庭娱乐电器联网 IEEE1394、以太网等,无线系统包括家庭无线网络协议 Home RF、蓝牙协议 Bluetooth 和 IEEE802.11 等。底层的协议、标准与所选择的传输介质有关,高层协议规范目前比较流行的有 HAVi、Jini、UPnP 等。

HAVi 是建立在 IEEE1394 底层协议的基础上,主要实现数字家电以及 HAVi 设备之间的数字音频/视频内容的传送及操作,如数字电视、数字摄像机、CD 等。互操作性是 HAVi 标准的主要特性,能够很方便地实现家庭内部不同娱乐设备之间的互相操作。在家庭娱乐系统中一般采用这种协议。

Jini 技术独立于具体的操作系统,组网简单、快捷和方便。它的运行环境基于 JVM (Java Virtual Machine),是一种动态的自动化机制,采用基于对象的协议,能够很方便地将家庭内部不同娱乐设备、通信设备、计算机和各种软件组合成一个动态单一的分布式系统,支持即插即用。

UPnP 则基于 XML 的描述原则提供了一种直接、灵活的方式实现设备的功能,采用基于数字的通信协议,支持对等 (Peer to Peer) 模型结构和即插即用。它独立于底层的物理媒体,适用于多种编程语言和不同的操作系统,支持家庭内部所有信息家电和设备的联网。

4.3 实现智能家庭网络的要求

智能家庭网络的实现需要一种共同的协议和标准 API,但由于家用电器和设备及家庭网络的多样性,使所有的家用电器和设备在相同的操作系统、API 和网络协议下实现联网操作,目前还是不可能的,许多用于高速网络的协议完全不适用于低速网络,这给智能家庭网络的实现带来了很大的困难。如何提供一个兼容所有传输介质、各种标准、协议的面向应用的接口技术,在应用层提供方便统一的家庭使用界面,实现家庭网络的互操作性,是智能家庭网络实现的关键。要求具体如下:

(1) 满足家庭自动化和信息化等要求,使家中设备良好地实现因特网共享。

(2) 灵活组态,扩展性能好,支持多种网络协议

并互相兼容。

(3) 支持即插即用,用户界面友好,操作简便。

(4) 末端设备接口成本低,要有较高的性价比。

(5) 安全性和可靠性,系统运行稳定。

4.4 智能家庭网络的控制平台

为了实现以上要求,提出了智能家庭网络的控制平台,如图 6 所示,智能家庭网络的控制平台是一个独立的平台,必须具备两方面的接口:与外部的接口和与家庭内部的接口。与外部的接口采用光纤、同轴电缆、无线等方式,用户可以通过网络浏览器来远

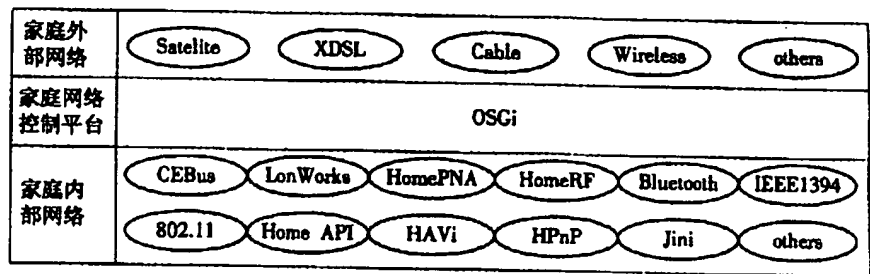


图 6 智能家庭网络控制平台

程访问它;与家庭内部的接口是指多个物理层接口,有 HomePNA、HomeRF、LonWorks、IEEE1394 接口等,通过这些接口,与家庭网络中的家电设备相联接。另外,必须实现不同网络高层协议之间的互连,以实现不同家庭网络之间的通信,如 HAVi、Jini 和 UPnP。OSGi (Open Service Gateway Initiative) 提供了一个开放的应用层和网关界面,能补充和提高所有的家庭网络标准和协议,包括 HomePNA、CEBus、Bluetooth、HomeRF、HPnP、Jini、IIAVi、HomeAPI 等;能管理各种网络设备、集成全部或部分已存在产品的控制机制,使所有出现在家庭网络平台上的协议和服务通过末端服务、住宅内部的电话线、同轴电缆、电线、双绞线等实现无缝的内部操作。OSGi 服务体系结构如图 7 所示。

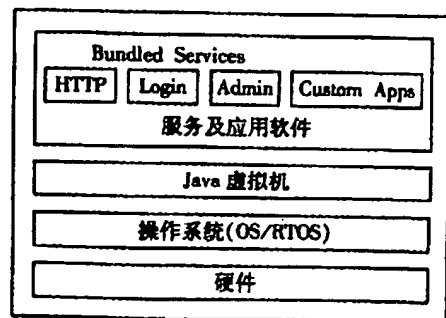


图 7 OSGi 服务体系结构

移动存储设备的技术特点和选用(一)

·综述·

周 宏¹, 马逢田¹, 刘昌明²

(1. 重庆工学院 工商管理系, 重庆 400050;

2. 重庆职业技术学院 重庆 400712)

移动存储设备是一类新兴的设备, 用于电脑存储和传送大量的文件和数据, 实现电脑之间的信息交换。随着移动存储应用需求的增大, 传统的软盘在速度、容量和可靠性等方面早已不能胜任, 大容量、轻巧便携、安全可靠、使用方便的移动存储设备在开始大量应用, 成为应用的热点。软驱作为电脑交换数据的设备, 是个人电脑发展史上变化最少的部件。随着电脑存储技术的进步, 估计软驱和软盘将在 2~3 年内被移动存储设备和其它存储设备全面取代。

移动存储设备一般有如下特点: 存储速度快、容量大; 存储数据安全可靠; 支持热插拔(带电插拔), 即插即用; 体积小、重量轻, 便于携带; 可用于大多数的电脑; 多数设备可不另接电源。

1 移动存储设备的接口

移动存储设备主要有 Flash 存储器、移动硬盘

[作者简介]

周宏, 教授, 管理科学与技术实验室主任。

刘昌明, 高级讲师。

马逢田, 实验室技术员。

采用智能家庭网络的控制平台, 其优点在于它提供一个可扩展的家庭网络构架, 消费者可以在不同的厂商之间选择产品, 而不需替换整个网络或网络体系的任何一部分。

5 展望

信息家电的出现, 为实现网络化住宅提供了基本条件; 嵌入式系统促进了信息家电、网络技术的迅速发展, 家庭网络的主要设备将不再是 PC 机, 而是嵌入式系统。智能家庭网络控制平台, 为家庭内、外部网络的联接及内部网络之间信息家电和设备的联接, 提供了一个基础平台。随着相关技术的不断进步, 它必将向着具有高度智能化、高度灵活性和互操作性的方向发展, 智能化、网络化的家庭就在不远的

(即外置式硬盘)、MO、Zip 等。按接口分有 USB, IEEE1394, SCSI, Parallel(LPT)等几种接口。了解移动存储设备, 需要对有关的接口及其性能有所了解。

1.1 USB

USB(Universal Serial Bus, 通用串行总线)是新型接口标准, USB1.1 可以提供 12 Mbps 的传输率, 速度较快, 是串口的 100 多倍, 用于中等传输速度的外部设备。Windows98 推出后, USB 的应用日趋成熟, 其优势明显。

USB 接口不占用任何 COM 口资源, 只占用一个 IRQ 地址。它支持热插拔, 微机上的 USB 接口直接为设备提供工作电源, 多数的 USB 设备不需要另外供电。用 USB 接口, 理论上最多可连接 127 个 USB 设备, 其连接的方式也十分灵活, 既可以串行连接, 也可以用 Hub 把多个设备连接在一起, 再同微机的 USB 口相接。但在实际的应用测试中, 现在的 USB 接口串联 4 个以上的设备就会有不正常的现象, 尽管如此, 比之于传统串口或并口只能连接一台设备, 还是要好得多。除一般的标准 USB 接口外, 还有不对称的 5 针 USB 接口。

未来。

[参考文献]

- [1] Amitava Dutta-Roy. Networks for Homes. IEEE SPECTRUM, 1999(12): 26-33.
- [2] Bill Rose. Home Networks: a Standards Perspective. IEEE Communications Magazine, 2001(12): 78-85.
- [3] Glenn T. Edens. Home Networking and CableHome Project at CableLabs. IEEE Communications Magazine, 2001(6): 112-121.
- [4] Li Gong. A Software Architecture for Open Service Gateways. IEEE Internet Computing, 2001(1), (2): 64-70.
- [5] <http://www.osgi.org>.
- [6] 金永权. 嵌入式系统、智能家电及家居网络. 单片机与嵌入式系统应用. 2001(1): 12-15.

[收稿日期] 2002-09-13